SQuAD : 100,000+ Questions for Machine Comprehension of Text

CS613 - Natural Language Processing

TEAM - 1

Bhavik Patel (22110047)

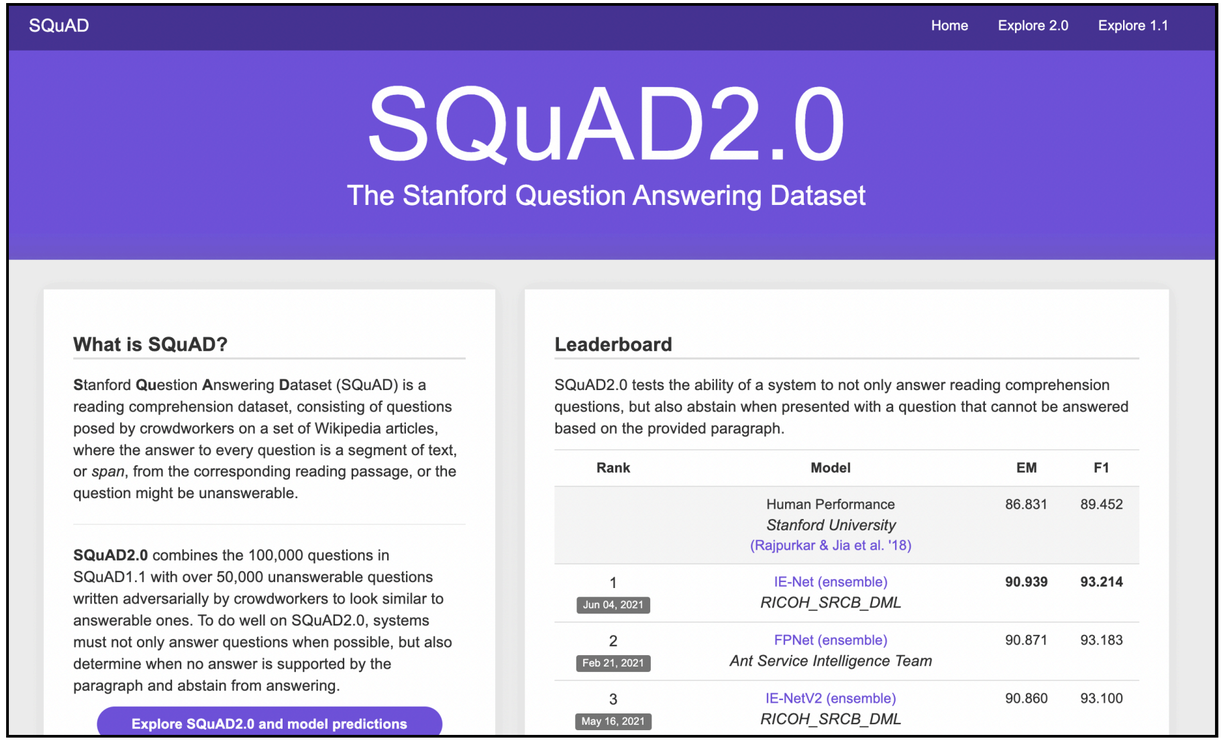
Guntas Singh Saran (22110089)

Hitesh Kumar (22110098)

Ruchit Jagodara (22110102)

Jinil Patel (22110184)

Indian Institue of Technology Gandhinagar
Palaj, Gujarat - 382355



SQuAD Dataset Homepage
(The Standford Question Answering Dataset)

Source: https://rajpurkar.github.io/SQuAD-explorer/

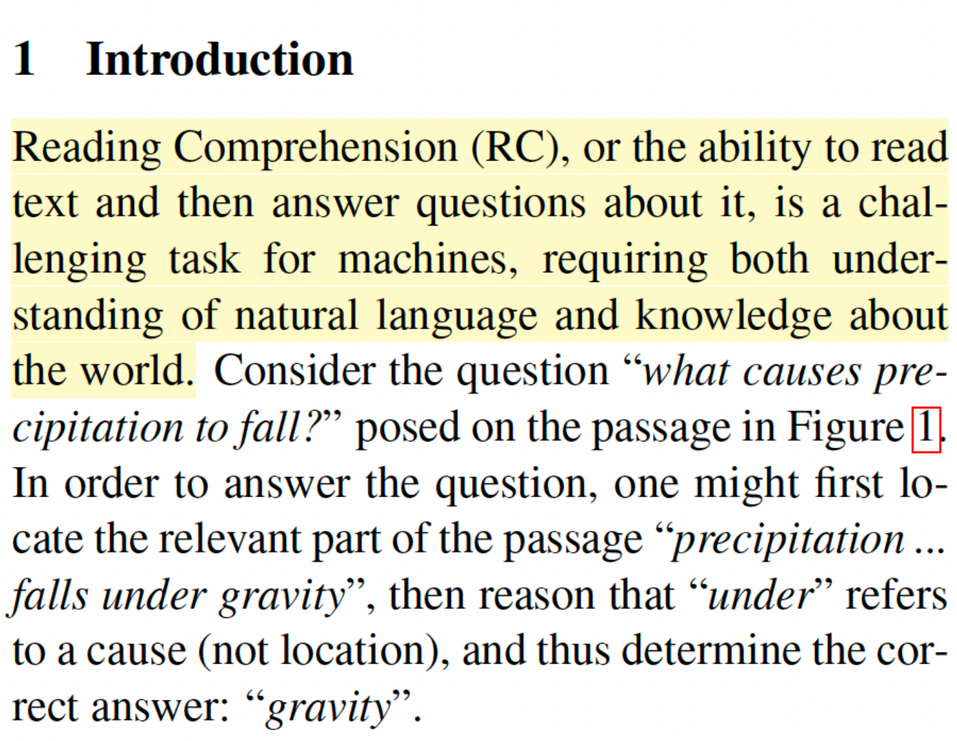
Reading Comprehension Task

What is the Reading Comprehension Task?

The reading comprehension task contains below steps:-
1. Read and understand the **comprehension / documents**.
2. Understand the **query / question** asked.
3. Find the **relevant information** (a document, a sentence, or a span) from the comprehension.

But there exist similar tasks similar to this has already been tackled like:-

1. **MCTest**, a very challenging task, was tackled by **Narasimhan and Barzilay,** (2015), **Sachan** et al. (2015).
2. **WikiQA** dataset (**Yang** et al., 2015), like SQuAD, uses Wikipedia passages as a source of answers
3. **CBT** (**Hill** et al., 2015) involves predicting a blanked-out workof a sentence given the 20 previous sentences .



1 Introduction

Reading Comprehension (RC), or the ability to read text and then answer questions about it, is a challenging task for machines, requiring both understanding of natural language and knowledge about the world. Consider the question "*what causes precipitation to fall?*" posed on the passage in Figure 1. In order to answer the question, one might first locate the relevant part of the passage "*precipitation ... falls under gravity*", then reason that "*under*" refers to a cause (not location), and thus determine the correct answer: "*gravity*".

Reading Comprehension Task

SQuAD v1.1

Warsaw

The Stanford Question Answering Dataset

One of the most famous people born in Warsaw was Maria Skłodowska-Curie, who achieved international recognition for her research on radioactivity and was the first female recipient of the Nobel Prize. Famous musicians include Władysław Szpilman and Frédéric Chopin. Though Chopin was born in the village of Żelazowa Wola, about 60 km (37 mi) from Warsaw, he moved to the city with his family when he was seven months old. Casimir Pulaski, a Polish general and hero of the American Revolutionary War, was born here in 1745.

What was Maria Curie the first female recipient of?
Ground Truth Answers: Nobel Prize | Nobel Prize | Nobel Prize
Prediction: Nobel Prize

What year was Casimir Pulaski born in Warsaw?
Ground Truth Answers: 1745 | 1745 | 1745
Prediction: 1745

Who was one of the most famous people born in Warsaw?
Ground Truth Answers: Maria Skłodowska-Curie | Maria Skłodowska-Curie | Maria Skłodowska-Curie
Prediction: Maria Skłodowska-Curie

Who was Frédéric Chopin?
Ground Truth Answers: Famous musicians | musicians | Famous musicians
Prediction: Władysław Szpilman

How old was Chopin when he moved to Warsaw with his family?
Ground Truth Answers: seven months old | seven months old | seven months old

Existing Similar Datasets

Richardson et. al. (2013): MCTest

Reading comprehension. A data-driven approach to reading comprehension goes back to Hirschman et al. (1999), who curated a dataset of 600 real 3rd–6th grade reading comprehension questions. Their pattern matching baseline was subsequently improved by a rule-based system (Riloff and Thelen, 2000) and a logistic regression model (Ng et al., 2000). More recently, Richardson et al. (2013) curated MCTest, which contains 660 stories created by crowdworkers, with 4 questions per story and 4 answer choices per question. Because many of the questions require commonsense reasoning and reasoning across multiple sentences, the dataset remains quite challenging, though there has been noticeable progress (Narasimhan and Barzilay, 2015; Sachan et al., 2015; Wang et al., 2015). Both curated datasets, although real and difficult, are too small to support very expressive statistical models.

James the Turtle was always getting in trouble. Sometimes he'd reach into the freezer and empty out all the food. Other times he'd sled on the deck and get a splinter. His aunt Jane tried as hard as she could to keep him out of trouble, but he was sneaky and got into lots of trouble behind her back.

One day, James thought he would go into town and see what kind of trouble he could get into. He went to the grocery store and pulled all the pudding off the shelves and ate two jars. Then he walked to the fast food restaurant and ordered 15 bags of fries. He didn't pay, and instead headed home.

His aunt was waiting for him in his room. She told James that she loved him, but he would have to start acting like a well-behaved turtle.

After about a month, and after getting into lots of trouble, James finally made up his mind to be a better turtle.

1) What is the name of the trouble making turtle?
A) Fries
B) Pudding
C) James
D) Jane

2) What did James pull off of the shelves in the grocery store?
A) pudding
B) fries
C) food
D) splinters

3) Where did James go after he went to the grocery store?
A) his deck
B) his freezer
C) a fast food restaurant
D) his room

4) What did James do after he ordered the fries?
A) went to the grocery store
B) went home without paying
C) ate them
D) made up his mind to be a better turtle

Figure 1. Sample Story and Questions (chosen randomly from MC500 train set).

MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text

Matthew Richardson
Microsoft Research
One Microsoft Way
Redmond, WA 98052
mattri@microsoft.com

Christopher J.C. Burges
Microsoft Research
One Microsoft Way
Redmond, WA 98052
cburges@microsoft.com

Erin Renshaw
Microsoft Research
One Microsoft Way
Redmond, WA 98052
erinren@microsoft.com

Existing Similar Datasets

Hosseini et. al. (2014)

Learning to Solve Arithmetic Word Problems with Verb Categorization

Mohammad Javad Hosseini[1], **Hannaneh Hajishirzi**[1], **Oren Etzioni**[2], **and Nate Kushman**[3]

[1]{hosseini, hannaneh}@washington.edu, [2]OrenE@allenai.org, [3]nkushman@csail.mit.edu
[1]University of Washington, [2]Allen Institute for AI, [3]Massachusetts Institute of Technology

Abstract

This paper presents a novel approach to learning to solve simple arithmetic word problems. Our system, ARIS, analyzes each of the sentences in the problem statement to identify the relevant variables and their values. ARIS then maps this information into an equation that represents the problem, and enables its (trivial) solution as shown in Figure 1. The paper analyzes the arithmetic-word problems "genre", identifying seven categories of verbs used in such problems. ARIS learns to categorize verbs with 81.2% accuracy, and is able to solve 77.7% of the problems in a corpus of standard primary school test questions. We report the first learning results on this task without reliance on predefined templates and make our data publicly available.[1]

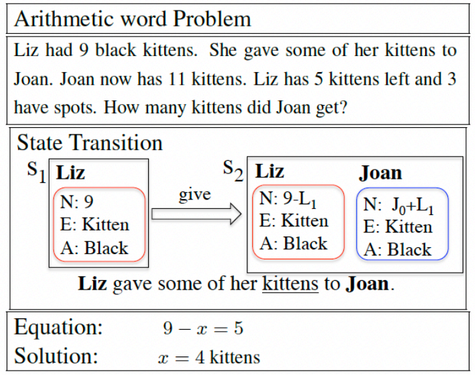


Figure 1: Example problem and solution.

make sense of multiple sentences, as shown in Figure 2, without *a priori* restrictions on the syntax or vocabulary used to describe the problem. Figure 1 shows an example where ARIS is asked to infer how many kittens Joan received based on facts and constraints expressed in the text, and represented

Kushman et. al. (2014)

Learning to Automatically Solve Algebra Word Problems

Nate Kushman[†], **Yoav Artzi**[‡], **Luke Zettlemoyer**[‡], **and Regina Barzilay**[†]

[†] Computer Science and Articial Intelligence Laboratory, Massachusetts Institute of Technology
{nkushman, regina}@csail.mit.edu
[‡] Computer Science & Engineering, University of Washington
{yoav, lsz}@cs.washington.edu

Abstract

We present an approach for automatically learning to solve algebra word problems. Our algorithm reasons across sentence boundaries to construct and solve a system of linear equations, while simultaneously recovering an alignment of the variables and numbers in these equations to the problem text. The learning algorithm uses varied supervision, including either full equations or just the final answers. We evaluate performance on a newly gathered corpus of algebra word problems, demonstrating that the system can correctly answer almost 70% of the questions in the dataset. This is, to our knowledge, the first learning result for this task.

1 Introduction

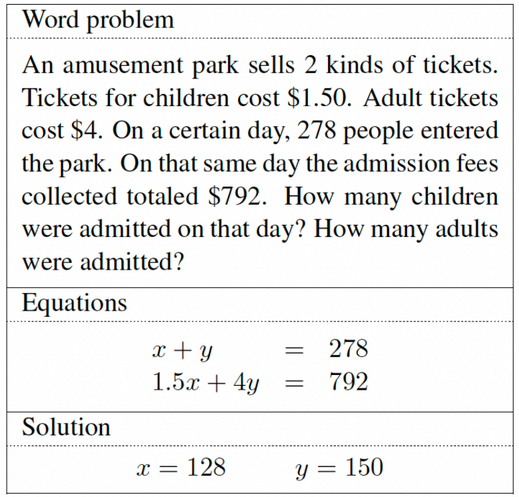


Figure 1: An example algebra word problem. Our goal is to map a given problem to a set of equations representing its algebraic meaning, which are then solved to get the problem's answer.

Existing Similar Datasets

Yang et. al. (2015): WikiQA

Open-domain question answering. The goal of open-domain QA is to answer a question from a large collection of documents. The annual evaluations at the Text REtreival Conference (TREC) (Voorhees and Tice, 2000) led to many advances in open-domain QA, many of which were used in IBM Watson for Jeopardy! (Ferrucci et al., 2013). Recently, Yang et al. (2015) created the WikiQA dataset, which, like SQuAD, use Wikipedia passages as a source of answers, but their task is sentence selection, while ours requires selecting a specific span in the sentence.

Question: Who wrote second Corinthians?

Second Epistle to the Corinthians The Second Epistle to the Corinthians, often referred to as Second Corinthians (and written as 2 Corinthians), is the eighth book of the New Testament of the Bible. Paul the Apostle and "Timothy our brother" wrote this epistle to "the church of God which is at Corinth, with all the saints which are in all Achaia".

Figure 1: An example question and the summary paragraph of a Wikipedia page.

WIKIQA: A Challenge Dataset for Open-Domain Question Answering

Yi Yang* Georgia Institute of Technology Atlanta, GA 30308, USA yiyang@gatech.edu

Wen-tau Yih **Christopher Meek** Microsoft Research Redmond, WA 98052, USA {scottyih,meek}@microsoft.com

Their task was **sentence selection** rather than **specific span** in the sentence

Existing Similar Datasets

Hermann et. al. (2015): Teaching Machines to Read and Comprehend

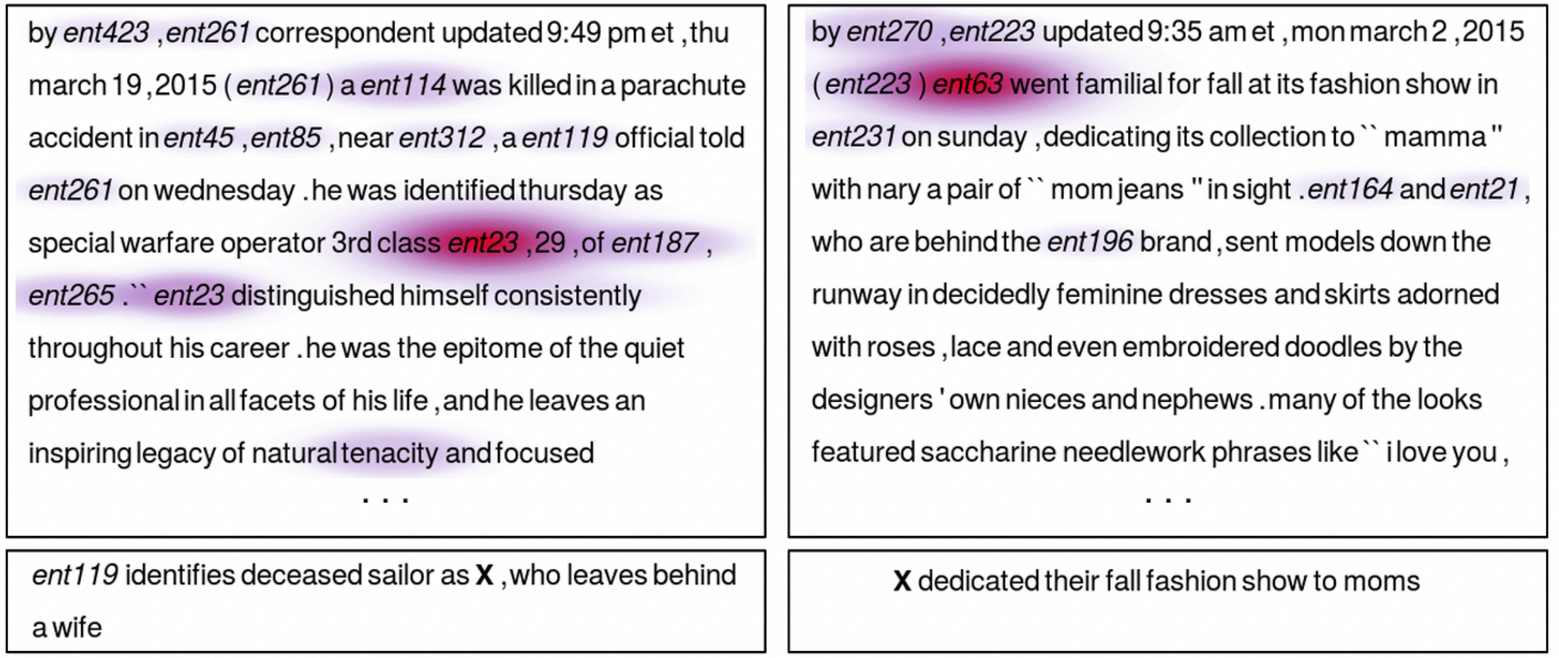


Figure 3: Attention heat maps from the Attentive Reader for two correctly answered validation set queries (the correct answers are *ent23* and *ent63*, respectively). Both examples require significant lexical generalisation and co-reference resolution in order to be answered correctly by a given model.

Cloze datasets. Recently, researchers have constructed cloze datasets, in which the goal is to predict the missing word (often a named entity) in a passage. Since these datasets can be automatically generated from naturally occurring data, they can be extremely large. The Children's Book Test (CBT) (Hill et al., 2015), for example, involves predicting a blanked-out word of a sentence given the 20 previous sentences. Hermann et al. (2015) constructed a corpus of cloze style questions by blanking out entities in abstractive summaries of CNN / Daily News articles; the goal is to fill in the entity based on the original article. While the size of this dataset is impressive, Chen et al. (2016) showed that the dataset requires less reasoning than previously thought, and concluded that performance is almost saturated.

One difference between SQuAD questions and cloze-style queries is that answers to cloze queries are single words or entities, while answers in SQuAD often include non-entities and can be much longer phrases. Another difference is that SQuAD focuses on questions whose answers are entailed by the passage, whereas the answers to cloze-style queries are merely suggested by the passage.

SQuAD v1.0

1. SQuAD **does not provide a list** of answer choices for each question.

2. Systems must **select the answer from all possible spans** in the passage, thus needing to cope with a fairly large number of candidates.

3. The team develop automatic techniques based on distances in **dependency trees** to quantify this **diversity** and **stratify** the questions by difficulty.

4. SQuAD contains **107,785 question-answer pairs** on **536 articles**, and is almost two orders of magnitude larger than previous manually labeled RC datasets such as **MCTest (Richardson et al., 2013)**

Dataset	Question source	Formulation	Size
SQuAD	**crowdsourced**	**RC, spans in passage**	**100K**
MCTest (Richardson et al., 2013)	crowdsourced	RC, multiple choice	2640
Algebra (Kushman et al., 2014)	standardized tests	computation	514
Science (Clark and Etzioni, 2016)	standardized tests	reasoning, multiple choice	855
WikiQA (Yang et al., 2015)	query logs	IR, sentence selection	3047
TREC-QA (Voorhees and Tice, 2000)	query logs + human editor	IR, free form	1479
CNN/Daily Mail (Hermann et al., 2015)	summary + cloze	RC, fill in single entity	1.4M
CBT (Hill et al., 2015)	cloze	RC, fill in single word	688K

Dataset Collection

3 Dataset Collection

We collect our dataset in three stages: curating passages, crowdsourcing question-answers on those passages, and obtaining additional answers.

Passage curation. To retrieve high-quality articles, we used Project Nayuki's Wikipedia's internal PageRanks to obtain the top 10000 articles of English Wikipedia, from which we sampled 536 articles uniformly at random. From each of these articles, we extracted individual paragraphs, stripping away images, figures, tables, and discarding paragraphs shorter than 500 characters. The result was 23,215 paragraphs for the 536 articles covering a wide range of topics, from musical celebrities to abstract concepts. We partitioned the articles randomly into a training set (80%), a development set (10%),

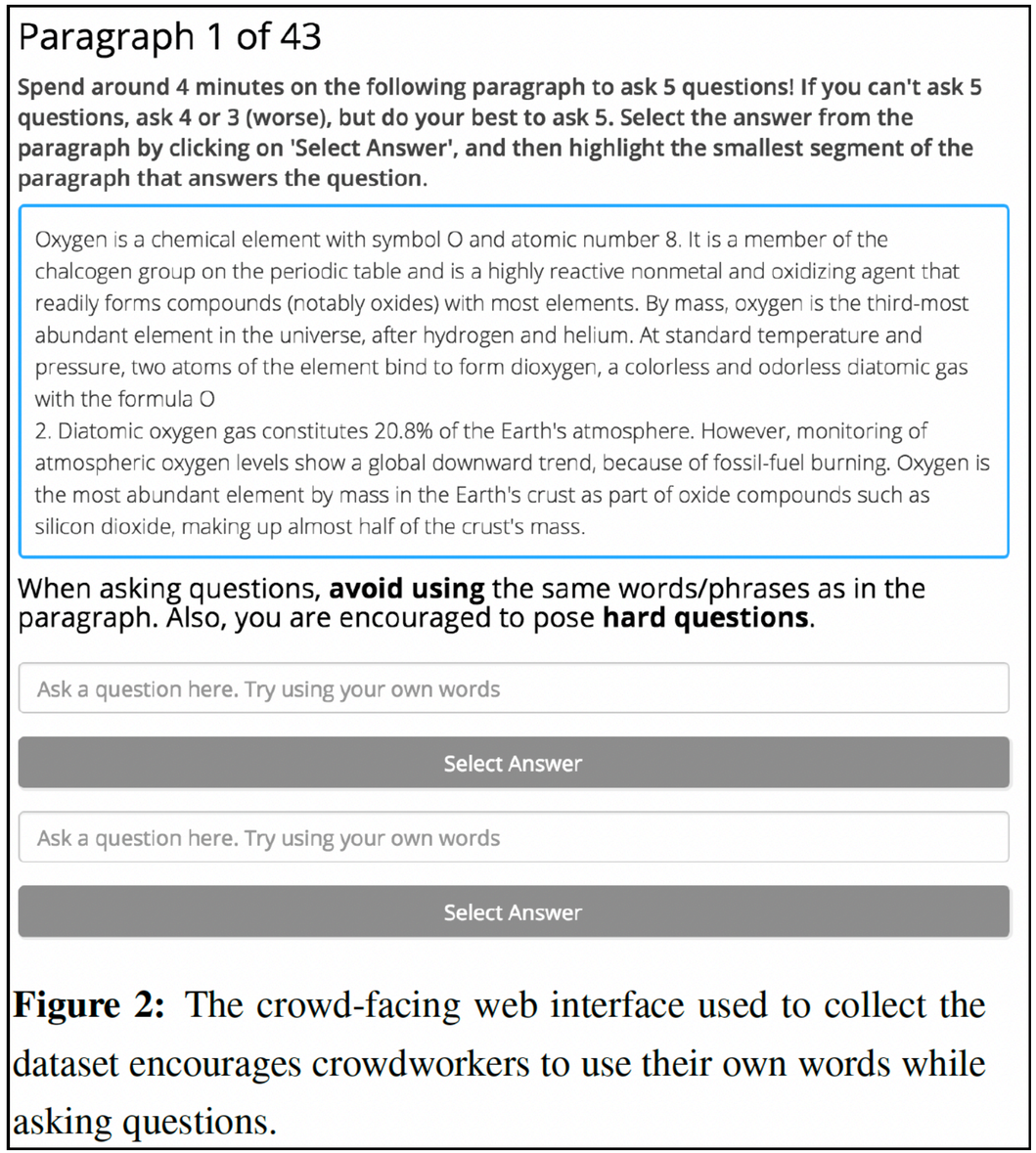


Figure 2: The crowd-facing web interface used to collect the dataset encourages crowdworkers to use their own words while asking questions.

Dataset Collection

1. Passage Curation: A total of 536 articles from English Wikipedia are sampled uniformly at random from the top 10,000 articles based on Project Nayuki's Wikipedia's internal PageRanks.

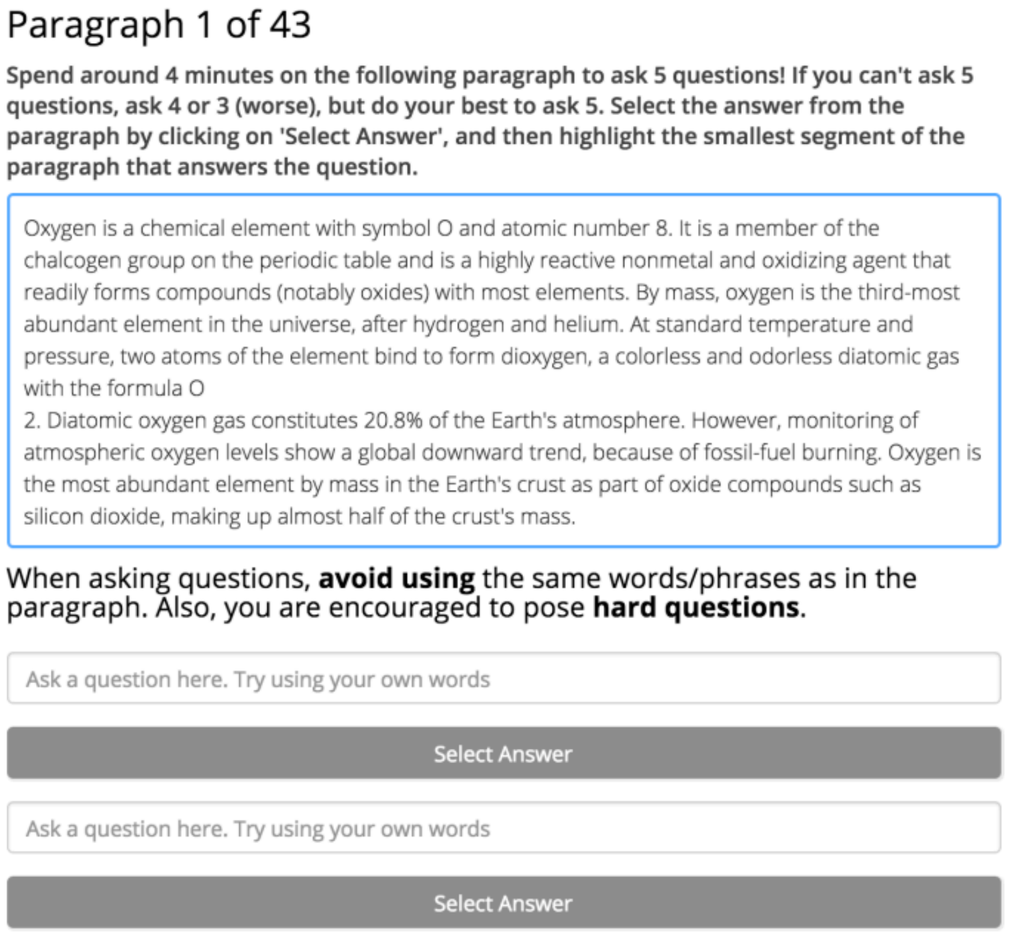
Training Set	Development Set	Test Set
80%	10%	10%

Dataset Collection

2. Question-Answer Collection:

Crowdworkers were employed to create questions. A 97% HIT acceptance rate was required, with 1000 HITs and US or Canada located.

A sample paragraph with good and bad questions and answers on that paragraph was also provided to guide the crowdworkers.



Dataset Collection

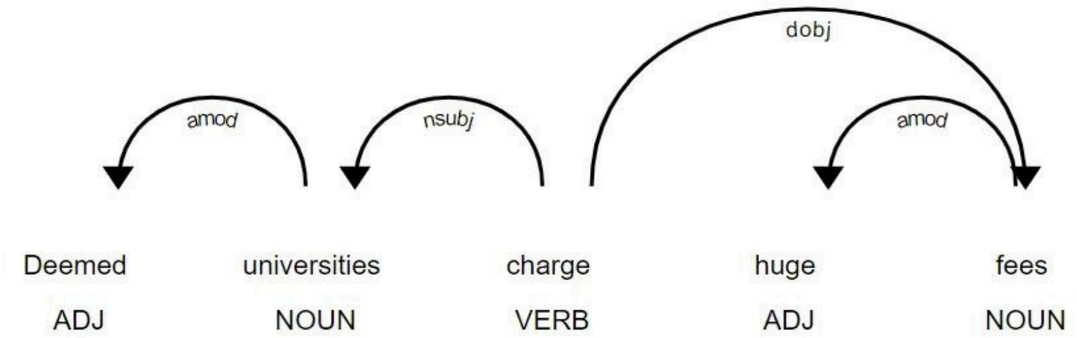
3. Additional Answers Collection:

- To **reduce penalties** for different but valid answers
- **Handling ambiguity** in badly framed questions
- **Variation** in question framing

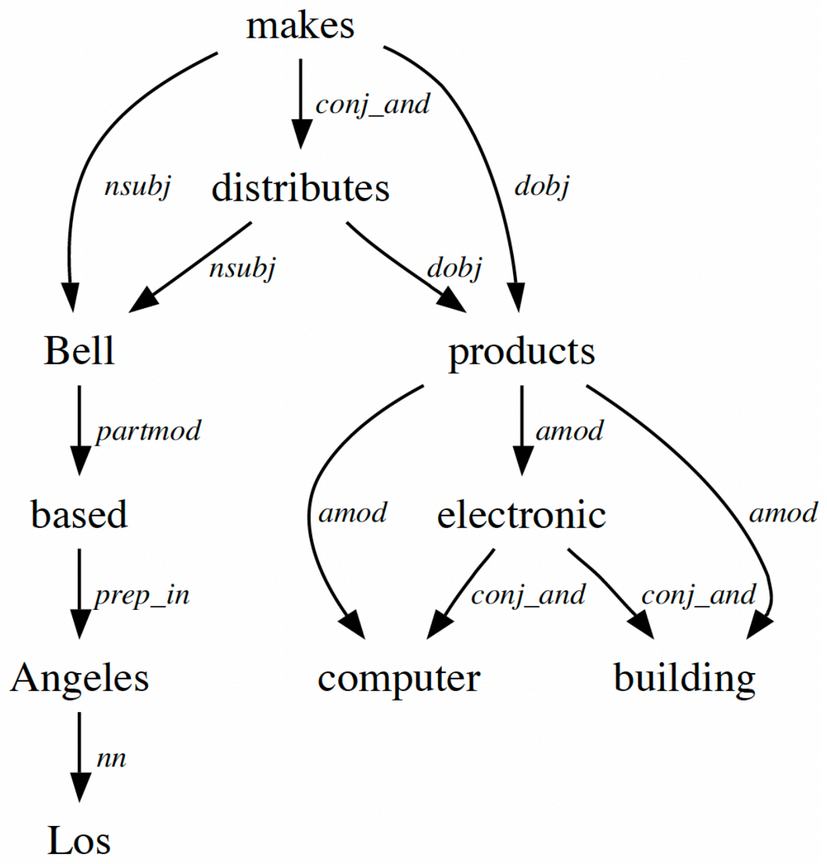
Dataset Analysis

4 Dataset Analysis

To understand the properties of SQuAD, we analyze the questions and answers in the development set. Specifically, we explore the (i) diversity of answer types, (ii) the difficulty of questions in terms of type of reasoning required to answer them, and (iii) the degree of syntactic divergence between the question and answer sentences.



Bell, based in Los Angeles, makes and distributes electronic, computer and building products.

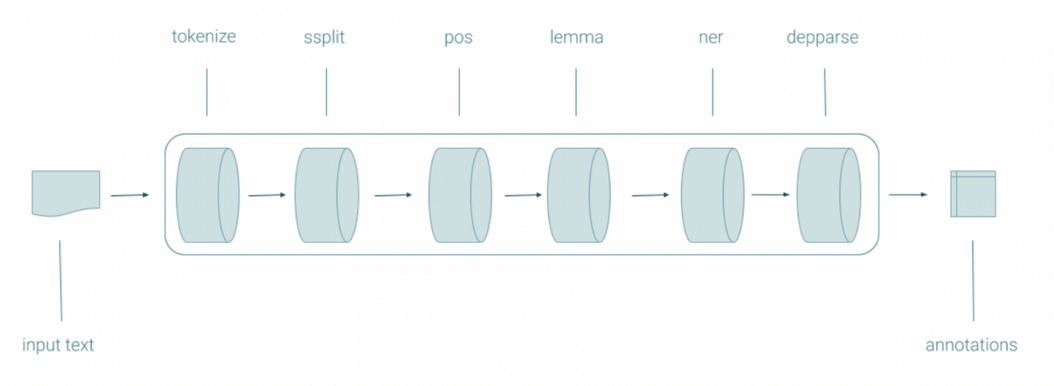


DEPENDENCY TREES GENERATED BY STANFORD CORENLP

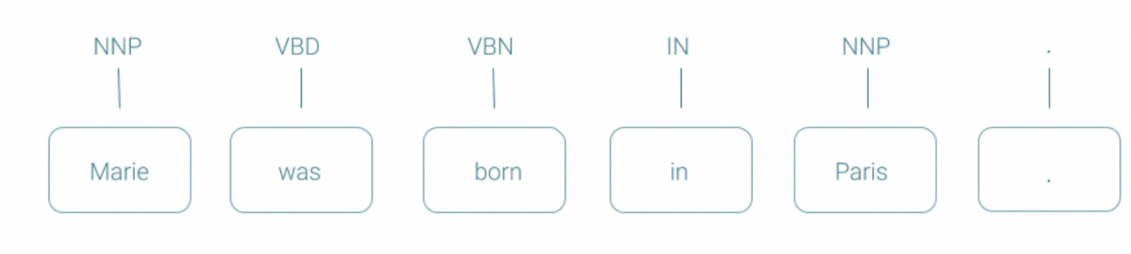
Source: Stanford typed dependencies manual Marie-Catherine de Marneffe and Christopher D. Manning

Stanford CoreNLP

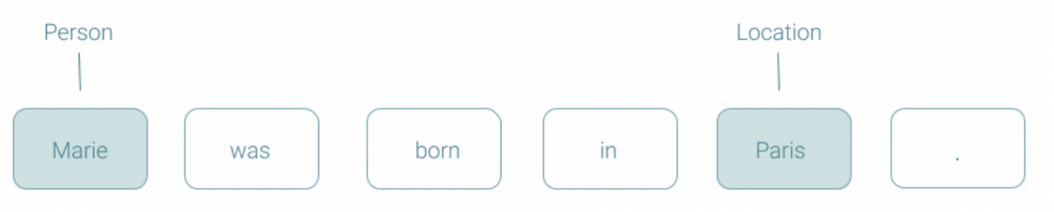
PIPELINE



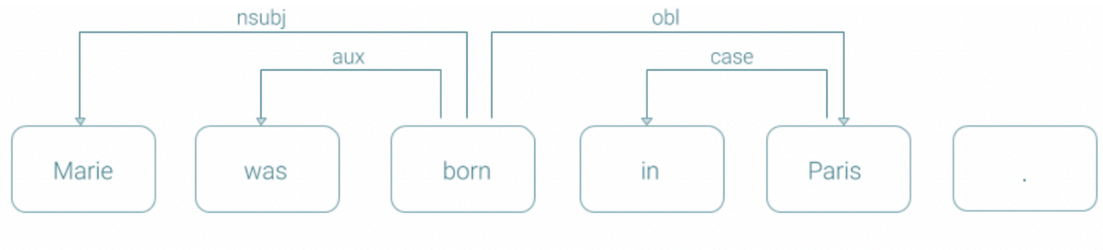
PART OF SPEECH



NAMED ENTITIES



DEPENDENCY TREES



Source: https://stanfordnlp.github.io/CoreNLP/

Dataset Analysis

Diversity in answers. We automatically categorize the answers as follows: We first separate the numerical and non-numerical answers. The non-numerical answers are categorized using constituency parses and POS tags generated by Stanford CoreNLP. The proper noun phrases are further split into person, location and other entities using NER tags. In Table 2, we can see dates and other numbers make up 19.8% of the data; 32.6% of the answers are proper nouns of three different types; 31.8% are common noun phrases answers; and the remaining 15.8% are made up of adjective phrases, verb phrases, clauses and other types.

Answer type	Percentage	Example
Date	8.9%	19 October 1512
Other Numeric	10.9%	12
Person	12.9%	Thomas Coke
Location	4.4%	Germany
Other Entity	15.3%	ABC Sports
Common Noun Phrase	31.8%	property damage
Adjective Phrase	3.9%	second-largest
Verb Phrase	5.5%	returned to Earth
Clause	3.7%	to avoid trivialization
Other	2.7%	quietly

Table 2: We automatically partition our answers into the following categories. Our dataset consists of large number of answers beyond proper noun entities.

Dataset Analysis

Reasoning required to answer questions. To get a better understanding of the reasoning required to answer the questions, we sampled 4 questions from each of the 48 articles in the development set, and then manually labeled the examples with the categories shown in Table 3. The results show that all examples have some sort of lexical or syntactic divergence between the question and the answer in the passage. Note that some examples fall into more than one category.

Reasoning	Description	Example	Percentage
Lexical variation (synonymy)	Major correspondences between the question and the answer sentence are synonyms.	Q: What is the Rankine cycle sometimes **called**? Sentence: The Rankine cycle is sometimes **referred** to as a practical Carnot cycle.	33.3%
Lexical variation (world knowledge)	Major correspondences between the question and the answer sentence require world knowledge to resolve.	Q: Which **governing bodies** have veto power? Sen.: **The European Parliament and the Council of the European Union** have powers of amendment and veto during the legislative process.	9.1%
Syntactic variation	After the question is paraphrased into declarative form, its syntactic dependency structure does not match that of the answer sentence even after local modifications.	Q: What Shakespeare scholar **is currently on the faculty**? Sen.: **Current faculty include** the anthropologist Marshall Sahlins, ..., Shakespeare scholar David Bevington.	64.1%
Multiple sentence reasoning	There is anaphora, or higher-level fusion of multiple sentences is required.	Q: What collection does **the V&A Theatre & Performance galleries** hold? Sen.: **The V&A Theatre & Performance galleries** opened in March 2009. ... **They** hold the UK's biggest national collection of material about live performance.	13.6%
Ambiguous	We don't agree with the crowdworkers' answer, or the question does not have a unique answer.	Q: What is the main goal of criminal punishment? Sen.: **Achieving crime control via incapacitation and deterrence** is a major goal of criminal punishment.	6.1%

Table 3: We manually labeled 192 examples into one or more of the above categories. Words relevant to the corresponding reasoning type are bolded, and the crowdsourced answer is underlined.

Dataset Analysis

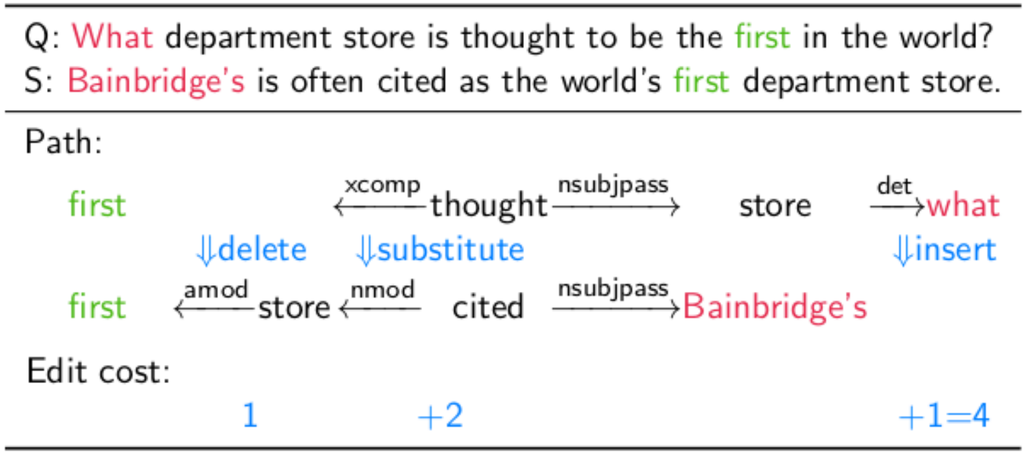


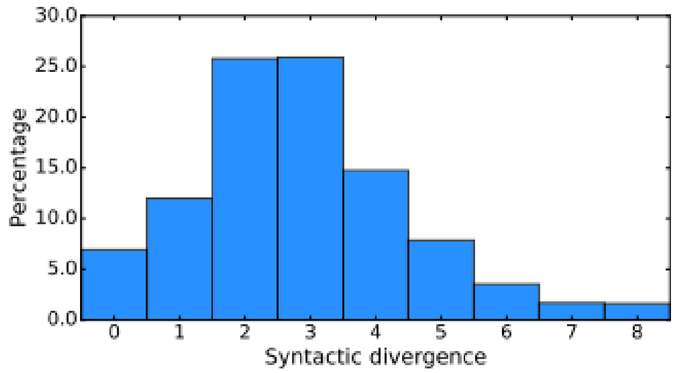
Figure 3: An example walking through the computation of the syntactic divergence between the question Q and answer sentence S.

Stratification by syntactic divergence. We also develop an automatic method to quantify the syntactic divergence between a question and the sentence containing the answer. This provides another way to measure the difficulty of a question and to stratify the dataset, which we return to in Section 6.3.

We illustrate how we measure the divergence with the example in Figure 3. We first detect anchors (word-lemma pairs common to both the question and answer sentences); in the example, the anchor is "*first*". The two unlexicalized paths, one from the anchor "*first*" in the question to the wh-word "*what*", and the other from the anchor in the answer sentence and to the answer span "*Bainbridge's*", are then extracted from the dependency parse trees. We measure the edit distance between these two paths, which we define as the minimum number of deletions or insertions to transform one path into the other. The syntactic divergence is then defined as the minimum edit distance over all possible anchors. The histogram in Figure 4a shows that there is a wide range of syntactic divergence in our dataset. We also show a concrete example where the edit distance is 0 and another where it is 6. Note that our syntactic divergence ignores lexical variation. Also, small divergence does not mean that a question is easy since there could be other candidates with similarly small divergence.

Dataset Analysis

Syntactic Divergence



(a) Histogram of syntactic divergence.

Q: Who went to Wittenberg to hear Luther speak?
S: Students thronged to Wittenberg to hear Luther speak.
Path:

Wittenberg $\xleftarrow{\text{nmod}}$ went $\xrightarrow{\text{nsubj}}$ Who
Wittenberg $\xleftarrow{\text{nmod}}$ thronged $\xrightarrow{\text{nsubj}}$ Students

(b) An example of a question-answer pair with edit distance 0 between the dependency paths (note that lexical variation is ignored in the computation of edit distance).

Q: What impact did the high school education movement have on the presence of skilled workers?
S: During the mass high school education movement from 1910 – 1940 , there was an increase in skilled workers.
Path:

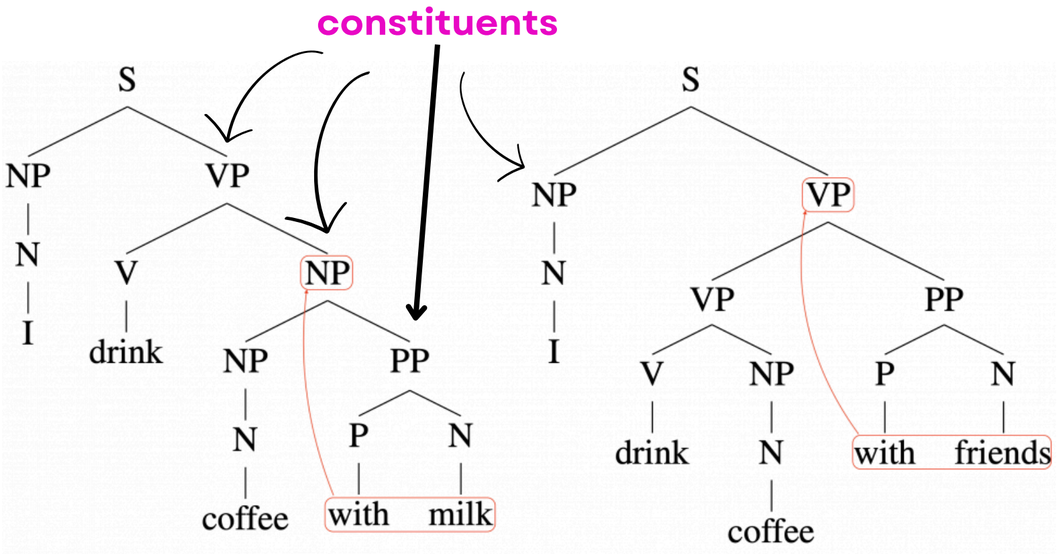
school $\xleftarrow{\text{compound}}$ movement $\xleftarrow{\text{nsubj}}$ have $\xrightarrow{\text{dobj}}$ impact $\xrightarrow{\text{det}}$ What
school $\xleftarrow{\text{compound}}$ movement $\xrightarrow{\text{nmod}}$ 1910 $\xrightarrow{\text{acl}}$ was $\xrightarrow{\text{nsubj}}$ increase

(c) An example of a question-answer pair with edit distance 6.

Figure 4: We use the edit distance between the unlexicalized dependency paths in the question and the sentence containing the answer to measure *syntactic divergence*.

Methods

Candidate answer generation. For all four methods, rather than considering all $O(L^2)$ spans as candidate answers, where L is the number of words in the sentence, we only use spans which are constituents in the constituency parse generated by Stanford CoreNLP. Ignoring punctuation and articles, we find that 77.3% of the correct answers in the development set are constituents. This places an effective ceiling on the accuracy of our methods. During training, when the correct answer of an example is not a constituent, we use the shortest constituent containing the correct answer as the target.



Constituency Parsing Tree

Compare the above two sentences ***"I drink coffee with milk"*** and ***"I drink coffee with friends"***. They only differ at their very last words, but their parses differ at earlier places, too.

Image Source: https://blog.ml.cmu.edu/2021/09/17/the-limitations-of-limited-context-for-constituency-parsing/

Methods

Sliding Window Baseline extended to Dependency Parsing Edit Distance

5.1 Sliding Window Baseline

For each candidate answer, we compute the unigram/bigram overlap between the sentence containing it (excluding the candidate itself) and the question. We keep all the candidates that have the maximal overlap. Among these, we select the best one using the sliding-window approach proposed in Richardson et al. (2013).

In addition to the basic sliding window approach, we also implemented the distance-based extension (Richardson et al., 2013). Whereas Richardson et al. (2013) used the entire passage as the context of an answer, we used only the sentence containing the candidate answer for efficiency.

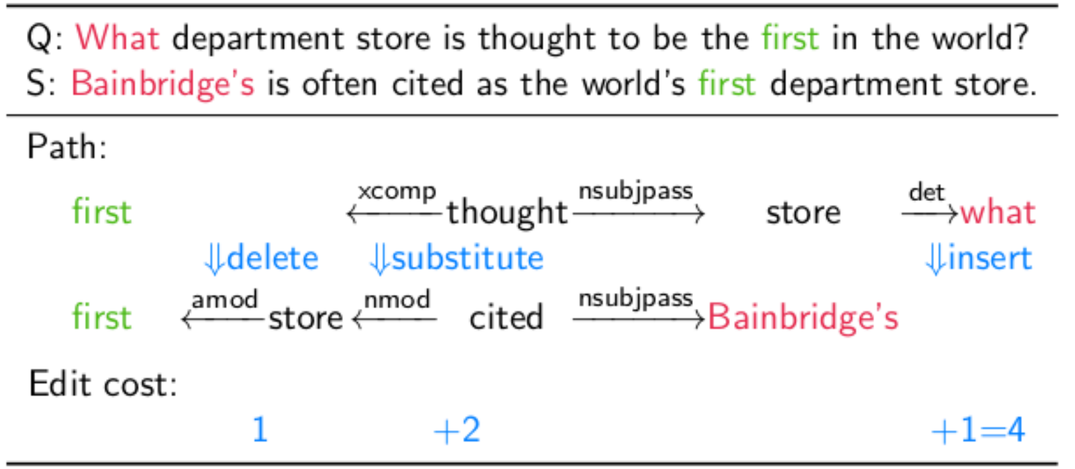


Figure 3: An example walking through the computation of the syntactic divergence between the question Q and answer sentence S.

Logistic Regression Model

5.2 Logistic Regression

In our logistic regression model, we extract several types of features for each candidate answer. We discretize each continuous feature into 10 equally-sized buckets, building a total of 180 million features, most of which are lexicalized features or dependency tree path features. The descriptions and examples of the features are summarized in Table 4.

The matching word and bigram frequencies as well as the root match features help the model pick the correct sentences. Length features bias the model towards picking common lengths and positions for answer spans, while span word frequencies bias the model against uninformative words. Constituent label and span POS tag features guide the model towards the correct answer types. In addition to these basic features, we resolve lexical variation using lexicalized features, and syntactic variation using dependency tree path features.

The multiclass log-likelihood loss is optimized using AdaGrad with an initial learning rate of 0.1. Each update is performed on the batch of all questions in a paragraph for efficiency, since they share the same candidates. L_2 regularization is used, with a coefficient of 0.1 divided by the number of batches. The model is trained with three passes over the train-

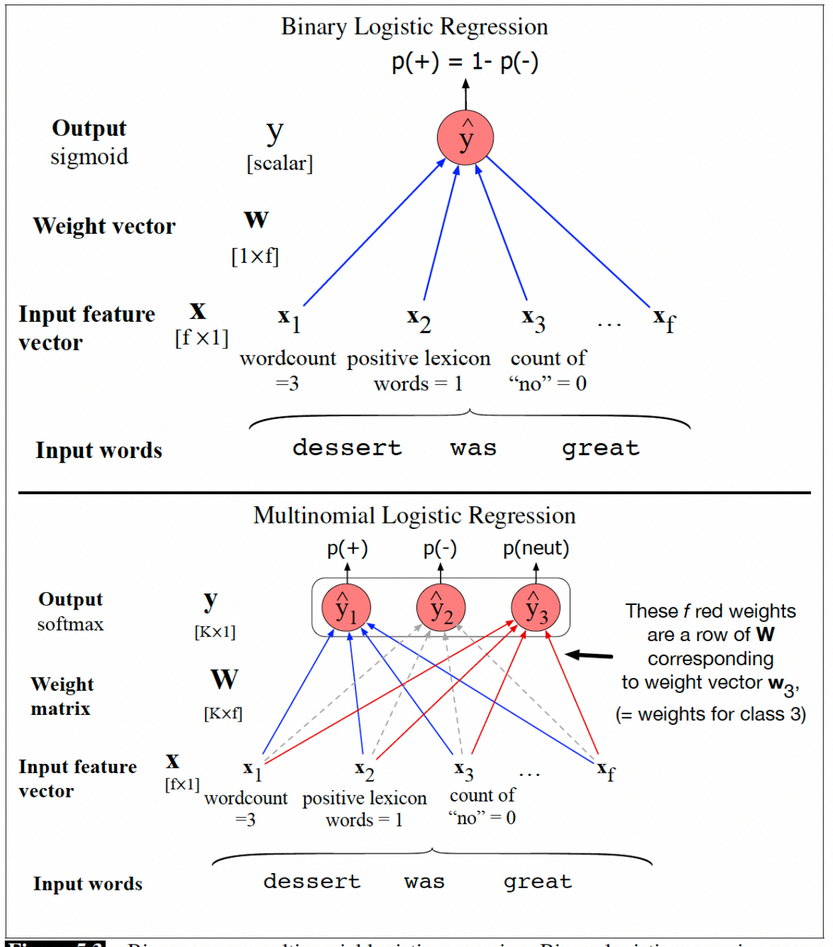


Figure 5.3 Binary versus multinomial logistic regression. Binary logistic regression uses a single weight vector $\mathbf{w}$, and has a scalar output $\hat{y}$. In multinomial logistic regression we have K separate weight vectors corresponding to the K classes, all packed into a single weight matrix $\mathbf{W}$, and a vector output $\hat{\mathbf{y}}$. We omit the biases from both figures for clarity.

Source: Speech and Language Processing. Daniel Jurafsky & James H. Martin.

Logistic Regression Model

Features used in the Logistic Regression Model

Feature Groups	Description	Examples
Matching Word Frequencies	Sum of the TF-IDF of the words that occur in both the question and the sentence containing the candidate answer. Separate features are used for the words to the left, to the right, inside the span, and in the whole sentence.	Span: $[0 \leq \text{sum} < 0.01]$ Left: $[7.9 \leq \text{sum} < 10.7]$
Matching Bigram Frequencies	Same as above, but using bigrams. We use the generalization of the TF-IDF described in Shirakawa et al. (2015).	Span: $[0 \leq \text{sum} < 2.4]$ Left: $[0 \leq \text{sum} < 2.7]$
Root Match	Whether the dependency parse tree roots of the question and sentence match, whether the sentence contains the root of the dependency parse tree of the question, and whether the question contains the root of the dependency parse tree of the sentence.	Root Match = False
Lengths	Number of words to the left, to the right, inside the span, and in the whole sentence.	Span: $[1 <= \text{num} < 2]$ Left: $[15 \leq \text{num} < 19]$
Span Word Frequencies	Sum of the TF-IDF of the words in the span, regardless of whether they appear in the question.	Span: $[5.2 \leq \text{sum} < 6.9]$
Constituent Label	Constituency parse tree label of the span, optionally combined with the wh-word in the question.	Span: NP Span: NP, wh-word: "*what*"
Span POS Tags	Sequence of the part-of-speech tags in the span, optionally combined with the wh-word in the question.	Span: [NN] Span: [NN], wh-word: "*what*"
Lexicalized	Lemmas of question words combined with the lemmas of words within distance 2 to the span in the sentence based on the dependency parse trees. Separately, question word lemmas combined with answer word lemmas.	Q: "*cause*", S: "*under*" $\xleftarrow{\text{case}}$ Q: "*fall*", A: "*gravity*"
Dependency Tree Paths	For each word that occurs in both the question and sentence, the path in the dependency parse tree from that word in the sentence to the span, optionally combined with the path from the wh-word to the word in the question. POS tags are included in the paths.	VBZ $\xrightarrow{\text{nmod}}$ NN what $\xleftarrow{\text{nsubj}}$ VBZ $\xrightarrow{\text{advcl}}$ + VBZ $\xrightarrow{\text{nmod}}$ NN

Table 4: Features used in the logistic regression model with examples for the question "*What causes precipitation to fall?*", sentence "*In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravity.*" and answer "*gravity*". Q denotes question, A denotes candidate answer, and S denotes sentence containing the candidate answer.

Logistic Regression Model

Example: Sentiment Analysis

Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon words $\in$ doc)	3
x_2	count(negative lexicon words $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	ln(word count of doc)	$\ln(66) = 4.19$

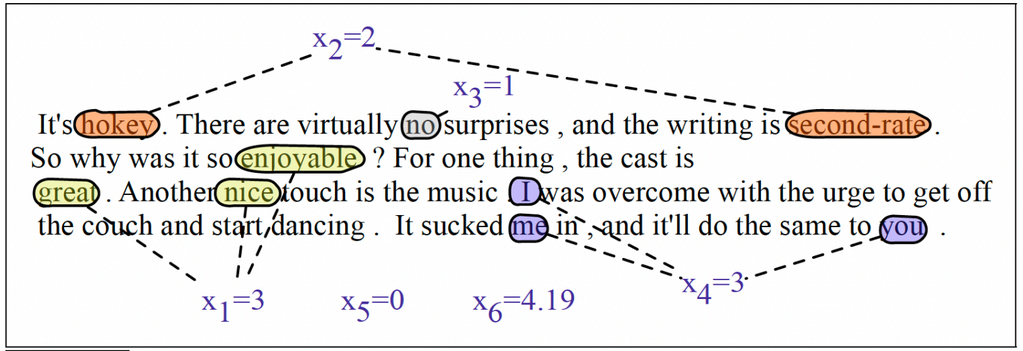


Figure 5.2 A sample mini test document showing the extracted features in the vector x.

Suppose we learned a real-valued weight for each of these features, and that the **6 weights** corresponding to the 6 features are **w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7]**, while **b = 0.1**

$$
\begin{aligned}
p(+|x) = P(y=1|x) &= \sigma(\mathbf{w}\cdot\mathbf{x}+b) \\
&= \sigma([2.5,-5.0,-1.2,0.5,2.0,0.7]\cdot[3,2,1,3,0,4.19]+0.1) \\
&= \sigma(.833) \\
&= 0.70 \qquad (5.8) \\
p(-|x) = P(y=0|x) &= 1-\sigma(\mathbf{w}\cdot\mathbf{x}+b) \\
&= 0.30
\end{aligned}
$$

Source: Speech and Language Processing. Daniel Jurafsky & James H. Martin.

Model Evaluation

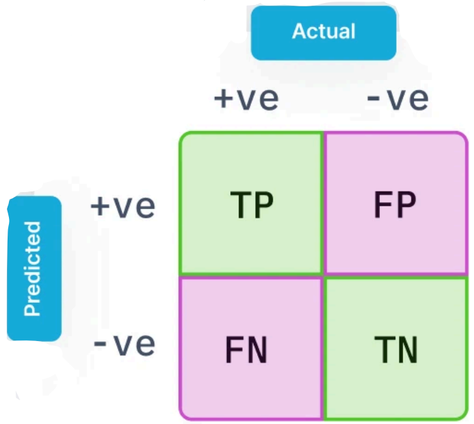
Performance Metrics

Exact match. This metric measures the percentage of predictions that match any one of the ground truth answers exactly.

(Macro-averaged) F1 score. This metric measures the average overlap between the prediction and ground truth answer. We treat the prediction and ground truth as bags of tokens, and compute their F1. We take the maximum F1 over all of the ground truth answers for a given question, and then average over all of the questions.

Model Evaluation

Performance Metrics

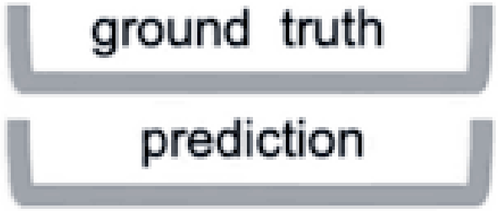


$$\text{Precision} = \frac{TP}{TP + FP}$$

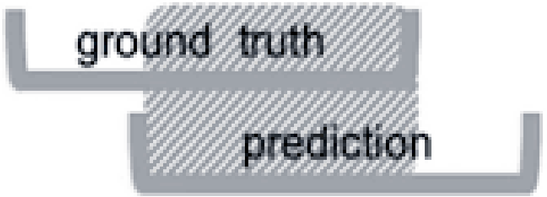
$$\text{Recall} = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Exact Match - Predicted answer exactly matches ground truth



F1 - Loose metrics, gives overlap between predicted answer and ground truth answer



Model Evaluation

Results on both Performance Measures

	Exact Match		F1	
	Dev	Test	Dev	Test
Random Guess	1.1%	1.3%	4.1%	4.3%
Sliding Window	13.2%	12.5%	20.2%	19.7%
Sliding Win. + Dist.	13.3%	13.0%	20.2%	20.0%
Logistic Regression	40.0%	40.4%	51.0%	51.0%
Human	80.3%	77.0%	90.5%	86.8%

Table 5: Performance of various methods and humans. Logistic regression outperforms the baselines, while there is still a significant gap between humans.

	F_1	
	Train	Dev
Logistic Regression	91.7%	51.0%
– Lex., – Dep. Paths	33.9%	35.8%
– Lexicalized	53.5%	45.4%
– Dep. Paths	91.4%	46.4%
– Match. Word Freq.	91.7%	48.1%
– Span POS Tags	91.7%	49.7%
– Match. Bigram Freq.	91.7%	50.3%
– Constituent Label	91.7%	50.4%
– Lengths	91.8%	50.5%
– Span Word Freq.	91.7%	50.5%
– Root Match	91.7%	50.6%

Table 6: Performance with feature ablations. We find that lexicalized and dependency tree path features are most important.

Model Evaluation

Comparison of Logistic Model with Human

	Logistic Regression Dev F1	Human Dev F1
Date	72.1%	93.9%
Other Numeric	62.5%	92.9%
Person	56.2%	95.4%
Location	55.4%	94.1%
Other Entity	52.2%	92.6%
Common Noun Phrase	46.5%	88.3%
Adjective Phrase	37.9%	86.8%
Verb Phrase	31.2%	82.4%
Clause	34.3%	84.5%
Other	34.8%	86.1%

Table 7: Performance stratified by answer types. Logistic regression performs better on certain types of answers, namely numbers and entities. On the other hand, human performance is more uniform.

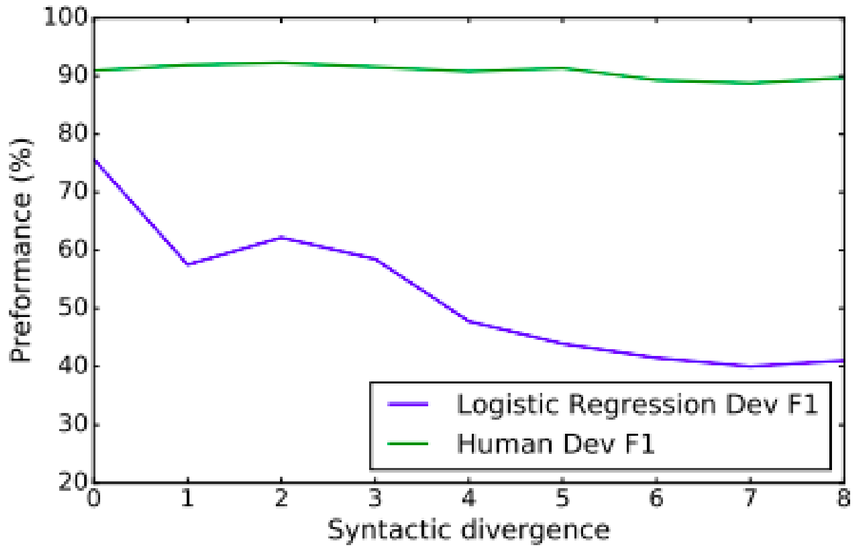


Figure 5: Performance stratified by syntactic divergence of questions and sentences. The performance of logistic regression degrades with increasing divergence. In contrast, human performance is stable across the full range of divergence.

Future Results

Wang & Jiang (2016): LSTM-based model on SQuAD v1.1

MACHINE COMPREHENSION USING MATCH-LSTM AND ANSWER POINTER

Shuohang Wang
School of Information Systems
Singapore Management University
shwang.2014@phdis.smu.edu.sg

Jing Jiang
School of Information Systems
Singapore Management University
jingjiang@smu.edu.sg

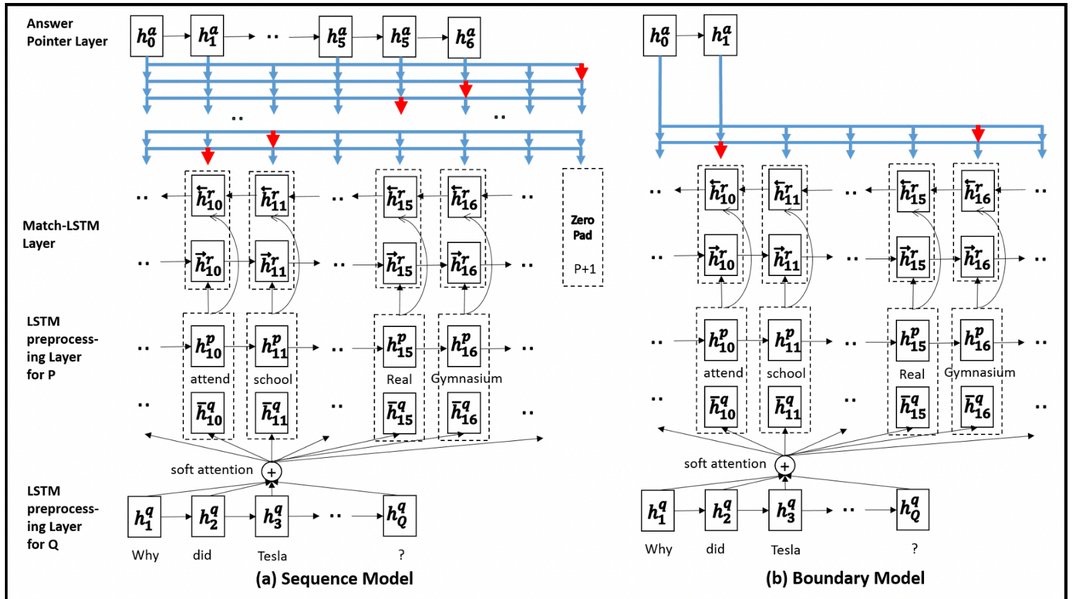


Figure 1: An overview of our two models. Both models consist of an LSTM preprocessing layer, a match-LSTM layer and an Answer Pointer layer. For each match-LSTM in a particular direction, $\bar{h}_i^q$, which is defined as $\mathbf{H}^q \alpha_i^\intercal$, is computed using the α in the corresponding direction, as described in either Eqn. (2) or Eqn. (5).

In 1870, Tesla moved to Karlovac, **to attend school at the Higher Real Gymnasium**, where he was profoundly influenced by a math teacher **Martin Sekulić**. The classes were held in **German**, as it was a school within the Austro-Hungarian Military Frontier. Tesla was able to perform integral calculus in his head, which prompted his teachers to believe that he was cheating. He finished a four-year term in three years, graduating in 1873.

1. In what language were the classes given?	German
2. Who was Tesla's main influence in Karlovac?	Martin Sekulić
3. Why did Tesla go to Karlovac?	attend school at the Higher Real Gymnasium

Table 1: A paragraph from Wikipedia and three associated questions together with their answers, taken from the SQuAD dataset. The tokens in bold in the paragraph are our predicted answers while the texts next to the questions are the ground truth answers.

Our contributions can be summarized as follows: (1) We propose two new end-to-end neural network models for machine comprehension, which combine match-LSTM and Ptr-Net to handle the special properties of the SQuAD dataset. (2) We have achieved the performance of an exact match score of 67.9% and an F1 score of 77.0% on the unseen test dataset, which is much better than the feature-engineered solution (Rajpurkar et al., 2016). Our performance is also close to the state of the art on SQuAD, which is 71.6% in terms of exact match and 80.4% in terms of F1 from Salesforce Research. (3) Our further analyses of the models reveal some useful insights for further improving the method. Beisdes, we also made our code available online [1].

Future Results

Rajpurkar et al. (2016): SQuAD 2.0

Know What You Don't Know: Unanswerable Questions for SQuAD

Pranav Rajpurkar* **Robin Jia*** **Percy Liang**
Computer Science Department, Stanford University
{pranavsr,robinjia,pliang}@cs.stanford.edu

Abstract

Extractive reading comprehension systems can often locate the correct answer to a question in a context document, but they also tend to make unreliable guesses on questions for which the correct answer is not stated in the context. Existing datasets either focus exclusively on answerable questions, or use automatically generated unanswerable questions that are easy to identify. To address these weaknesses, we present SQuAD 2.0, the latest version of the Stanford Question Answering Dataset (SQuAD). SQuAD 2.0 combines existing SQuAD data with over 50,000 unanswerable questions written adversarially by crowdworkers to look similar to answerable ones. To do well on SQuAD 2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering. SQuAD 2.0 is a challenging natural language understanding task for existing models: a strong neural system that gets 86% F1 on SQuAD 1.1 achieves only 66% F1 on SQuAD 2.0.

Article: Endangered Species Act
Paragraph: "... *Other legislation followed, including the Migratory Bird Conservation Act of 1929, a 1937 treaty prohibiting the hunting of right and gray whales, and the Bald Eagle Protection Act of 1940. These later laws had a low cost to society—the species were relatively rare—and little opposition was raised.*"

Question 1: "*Which laws faced significant opposition?*"
Plausible Answer: *later laws*

Question 2: "*What was the name of the 1937 treaty?*"
Plausible Answer: *Bald Eagle Protection Act*

Figure 1: Two unanswerable questions written by crowdworkers, along with plausible (but incorrect) answers. Relevant keywords are shown in blue.

even produced systems that surpass human-level exact match accuracy on the Stanford Question Answering Dataset (SQuAD), one of the most widely-used reading comprehension benchmarks (Rajpurkar et al., 2016).

Nonetheless, these systems are still far from true language understanding. Recent analysis shows that models can do well at SQuAD by learning context and type-matching heuristics (Weissenborn et al., 2017), and that success on SQuAD does not ensure robustness to distracting sen-

Leaderboard

SQuAD2.0 tests the ability of a system to not only answer reading comprehension questions, but also abstain when presented with a question that cannot be answered based on the provided paragraph.

Rank	Model	EM	F1
	Human Performance *Stanford University* (Rajpurkar & Jia et al. '18)	86.831	89.452
1 Jun 04, 2021	IE-Net (ensemble) *RICOH_SRCB_DML*	**90.939**	**93.214**
2 Feb 21, 2021	FPNet (ensemble) *Ant Service Intelligence Team*	90.871	93.183
3 May 16, 2021	IE-NetV2 (ensemble) *RICOH_SRCB_DML*	90.860	93.100
4 Apr 06, 2020	SA-Net on Albert (ensemble) *QIANXIN*	90.724	93.011
5 May 05, 2020	SA-Net-V2 (ensemble) *QIANXIN*	90.679	92.948
5 Apr 05, 2020	Retro-Reader (ensemble) *Shanghai Jiao Tong University* http://arxiv.org/abs/2001.09694	90.578	92.978
5 Feb 05, 2021	FPNet (ensemble) *YuYang*	90.600	92.899

Source: https://rajpurkar.github.io/SQuAD-explorer/

Future Results

This contributed to the "Oil Shock". After 1971, OPEC was slow to readjust prices to reflect this depreciation. From 1947 to 1967, the dollar price of oil had risen by less than two percent per year. Until the oil shock, the price had also remained fairly stable versus other currencies and commodities. OPEC ministers had not developed institutional mechanisms to update prices in sync with changing market conditions, so their real incomes lagged. The substantial price increases of 1973–1974 largely returned their prices and corresponding incomes to Bretton Woods levels in terms of commodities such as gold.

The price of oil is usually a stable commodity until when?
Ground Truth Answers: Until the oil shock | the oil shock | the oil shock | Until the oil shock | the oil shock
Prediction: oil shock

How much had the price of gold risen after 1971?
Ground Truth Answers: <No Answer>
Prediction: <No Answer>

Until the oil shock how had the price of gold been?
Ground Truth Answers: <No Answer>
Prediction: <No Answer>

What happened to Bretton Woods income due to prices not being in sync with the market?
Ground Truth Answers: <No Answer>
Prediction: <No Answer>

When did institutional mechanisms finally return to Bretton Woods levels?
Ground Truth Answers: <No Answer>
Prediction: <No Answer>

THANK YOU

TEAM - 1

Bhavik Patel (bhavik.patel@iitgn.ac.in)

Guntas Singh Saran (guntassingh.saran@iitgn.ac.in)

Hitesh Kumar (hitesh.kumar@iitgn.ac.in)

Ruchi Jagodara (ruchit.jagodara@iitgn.ac.in)

Jinil Patel (jinilkumar.patel@iitgn.ac.in)

Indian Institue of Technology Gandhinagar
Palaj, Gujarat - 382355